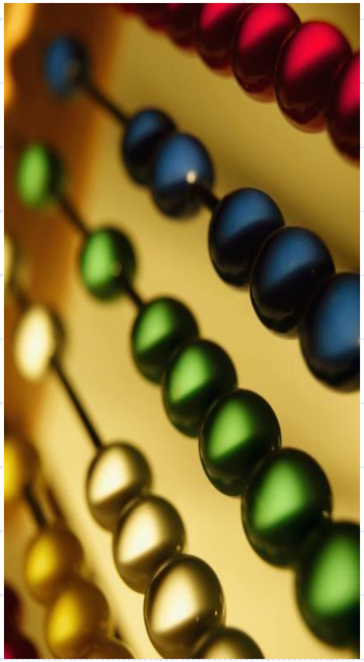
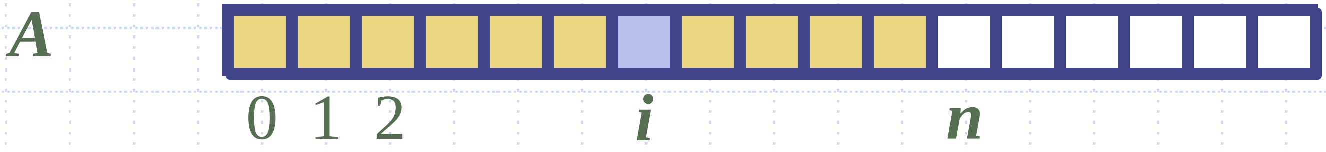
Arrays

Dr. Nadia M. Mohammed



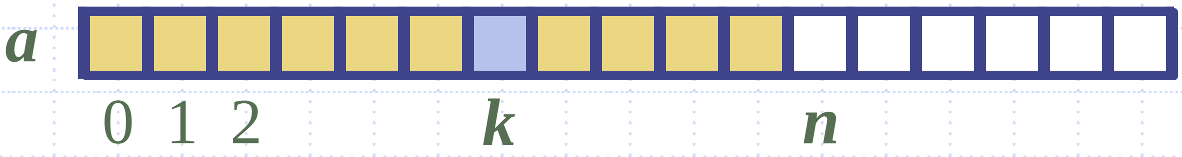
Array Definition

- An ***array*** is a sequenced collection of variables all of the same type. Each variable, or ***cell***, in an array has an ***index***, which uniquely refers to the value stored in that cell. The cells of an array, A, are numbered 0, 1, 2, and so on.
- Each value stored in an array is often called an ***element*** of that array.



Array Length and Capacity

- Since the length of an array determines the maximum number of things that can be stored in the array ---***capacity***.
  - numbered 0, 1, 2, and so on, up through *a*.length−1,
  - the cell with index *k* can be accessed with syntax *a*[*k*].

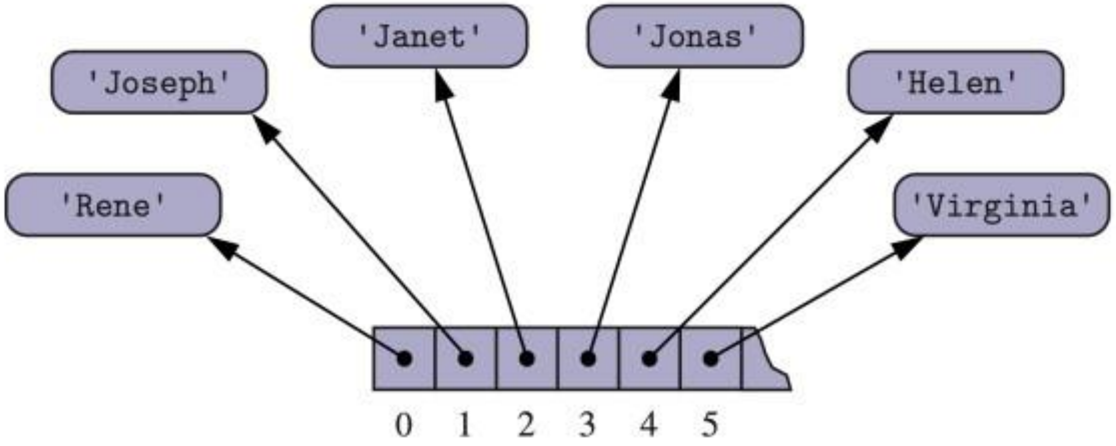


Arrays of Characters or Object References

- An array can store primitive elements, such as characters.

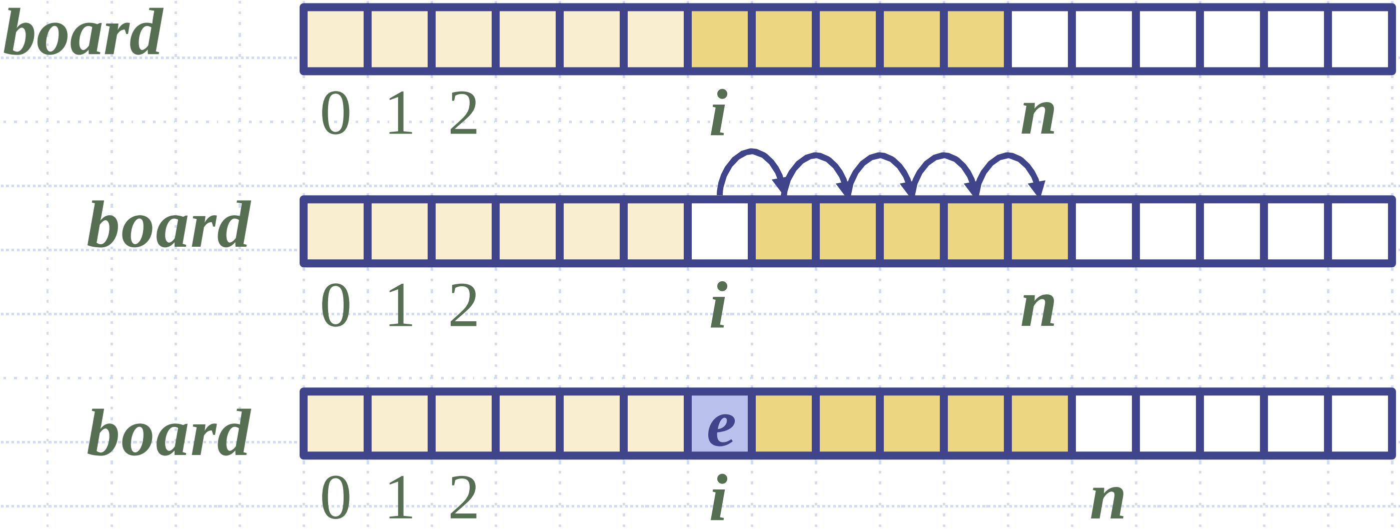
S	A	M	P	L	E
0	1	2	3	4	5

- An array can also store references to objects.



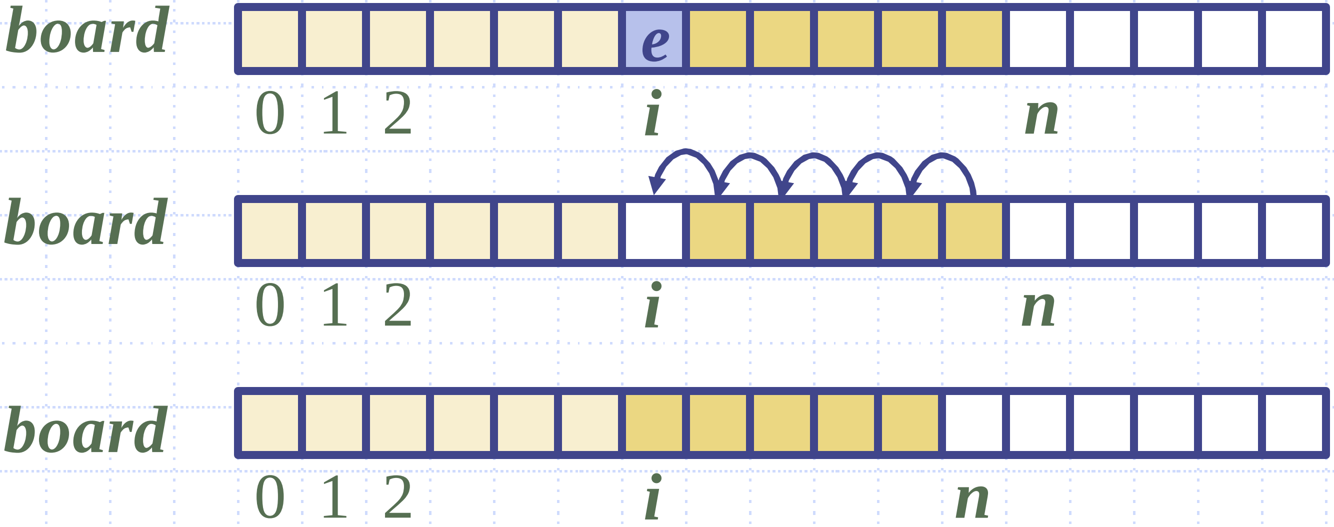
Adding an Entry

- To add an entry e into array board at index i, we need to make room for it by shifting forward the ***n* – *i*** entries ***board***[***i***], …, ***board***[***n*** – 1]



Removing an Entry

- To remove the entry e at index i, we need to fill the hole left by e by shifting backward the $n - i - 1$ elements $\boldsymbol{board}[\boldsymbol{i} + 1], \ldots, \boldsymbol{board}[\boldsymbol{n} - 1]$



Disadvantages of arrays?

- Need to know the maximum size before hand
- If elements need to be kept in order
  - Inserting/deleting an element requires moving (lots of) elements

Represented One Dim. Array in memory

- X : Array [1.. N] of integer {any other type}
- Location {X[I]} = Base Address + (I – 1)

- Example :calculate the address of element No. 4 in one dim. Array z: Array [1..6] of integer, where base address = 500
- Location {z(4)} =   500 + (4-1)
-                              = 503

Represented Two Dim. Array in memory Row Wise method

- A : Array [1.. M, 1..N] of integer {any other type}
- Location {A[I,J]} = Base Address + N * (I – 1) + (J-1)
- Where N:total number of column
- I: row number
- J:column Number
- Example :calculate the address of element T[4,6] in two dim. Array T: Array [1..5, 1..7] of integer, where base address = 900
- Location {T(4,6)} = 900 +7 * (4-1) + (6-1)
- = 926

Represented Two Dim. Array in memory Column Wise method

- A : Array [1.. M, 1..N] of integer {any other type}
- Location {A[I,J]} = Base Address + M * (J – 1) + (I-1)
- Where M:total number of row
- I: row number
- J:column Number
- Example :calculate the address of element T[5,7] in two dim. Array T: Array [1..6, 1..8] of integer, where base address = 300
- Location {T(5,7)} = 300 + 6 * (7-1) + (5-1)
- = 340